Amendments to the Claims:

This listing of claims will replace all prior versions, and listing of claims in the application:

Listing of the Claims:

1-7. (Cancelled)

8. (Cancelled)

9. (Cancelled)

10. (Currently Amended) ~~A method as recited in claim 8~~ In a Java™ computing environment, a method of identifying active Java™ objects and active Java™ classes by a virtual machine at runtime, said method comprising:

generating and loading in the virtual machine prior to execution time a cluster of Java™ object representations which are sequentially represented inside the virtual machine, wherein each of said Java™ object representations in said cluster consists of:

a first reference to an internal class representation of a class associated with a Java™ object, and

a second reference to instance fields associated with said Java™ object;

sequentially reading by said virtual machine at runtime said cluster of Java™ object representations;

determining by said virtual machine at runtime whether Java™ objects or Java™ classes are to be identified;

using said second references of said cluster to mark memory addresses that correspond to Java™ objects when said determining determines that Java™ objects are to be identified, thereby allowing Java™ objects to be identified at run time by a sequential read of said cluster; and

using one or more of said first references of said cluster to mark memory addresses that correspond to Java™ classes when said determining determines that Java™ classes are to be identified, thereby allowing Java™ classes to be identified at run time by a sequential read of said cluster, wherein said first reference is a direct reference to said internal class representation of said Java™ object.

11. (Currently Amended) A method as recited in claim 10,

wherein said second reference is a reference to an array of references, and

wherein each reference in said array of references is a reference to an instance field associated with said Java™ object.

12. (Previously Presented) A method as recited in claim 10, wherein said first and second references are allocated as four bytes.

13. (Previously Presented) A method as recited in claim 10, wherein said method further comprises:

removing internal class representations that have not been marked.

14. (Currently Amended) A method as recited in claim 10, wherein said method further comprises:

removing Java™ objects that have not been marked.

15. (Currently Amended) A method as recited in claim 10, wherein said method is used by a virtual machine for garbage collection of Java™ objects and Java™ classes.

16. (Cancelled)

17. (Cancelled)

18. (Currently Amended) ~~A computer readable medium as recited in claim 28~~ A computer readable medium including at least computer program code for identifying active Java™ objects and active Java™ classes by a virtual machine at runtime, comprising:

computer program code for generating and loading in the virtual machine prior to execution time a cluster of Java™ object representations which are sequentially represented inside the virtual machine, wherein each of said Java™ object representations in said cluster consists of:

a first reference to an internal class representation of a class associated with a Java™ object, and

a second reference to instance fields associated with said Java™ object;

computer program code for sequentially reading by said virtual machine at runtime said cluster of Java™ object representations~~.~~;

computer program code for determining by said virtual machine at runtime whether Java™ objects or Java™ classes are to be identified;

computer program code for using said second references of said cluster to mark memory addresses that correspond to Java™ objects when said determining determines that Java™ objects are to be identified, thereby allowing Java™ objects to be identified at run time by a sequential read of said cluster; and

computer program code for using one or more of said first references of said cluster to mark memory addresses that correspond to Java™ classes when said determining determines that Java™ classes are to be identified, thereby allowing Java™ classes to be identified at run time by a sequential read of said cluster, wherein said first reference is a direct reference to said internal class representation of said Java™ object.

19. (Currently Amended) A computer readable medium as recited in claim 18,

wherein said second reference is a reference to an array of references, and

wherein each reference in said array of references is a reference to an instance field associated with said Java™ object.

20. (Original) A computer readable medium as recited in claim 19, wherein said first and second references are allocated as four bytes.

21. (Cancelled)

22. (Cancelled)

23. (Cancelled)

24. (Cancelled)

25. (Currently Amended) A ~~virtual machine~~ computer-readable medium as recited in claim ~~21~~18, ~~wherein said method~~ further ~~comprises~~comprising:

computer program code for removing internal class representations that have not been marked.

26. (Currently Amended) A ~~virtual machine~~ computer-readable medium as recited in claim ~~21~~18, ~~wherein said method~~ further ~~comprises~~comprising:

computer program code for removing Java™ objects that have not been marked.

27. (Currently Amended) A ~~virtual machine~~ computer-readable medium as recited in claim ~~21~~18, wherein said Java™ objects are identified for garbage collection at runtime.

28. (Cancelled)